Adaptive and Extensible Virtualization for Exascale

Position Paper for the Workshop on Exascale Operating Systems and Runtime Software

Peter A. Dinda
Northwestern University
pdinda@northwestern.edu

July 12, 2012

1 Description

The community believes that existing node-level operating systems (OSes) will not be able to meet the challenges of parallelism, reliability, deep memory hierarchies, and power management in exascale systems. However, moving OS innovations from research into deployment has always been difficult. When these innovations require changes to applications, the difficulty is compounded.

In the data center and cloud computing contexts, similar difficulties have been at least partially surmounted through the introduction of a virtualization layer. Decoupling the hardware and the OS has allowed the continued use of current OSes and applications, while it has also enabled OS evolution that would not otherwise be possible. Furthermore, considerable innovation has centered around the virtual machine monitor (VMM) itself, which is, after all, a new OS, one that, other than providing a virtualized or paravirtualized hardware interface for backward compatibility, is a blank slate that enables innovation.

We should take advantage of the exascale transition to make the leap to a virtualization layer. This layer would open future exascale machines up to multiple uses, enhancing the business case for building them. It would also provide an incremental adoption path for existing HPC applications. The focus here, however, is on how such a layer might directly and indirectly address the specific challenges described above.

Let us consider the virtualization layer as consisting of one VMM managing each node, with these VMMs coordinating their course-grained actions across the machine as a whole. This virtualization layer would have two important properties: it would be adaptive and extensible.

Adaptivity An adaptive virtualization layer would continuously make node-local and machine-scale decisions that attempted to optimize the execution environment with respect to performance, reliability, and power/energy. Specific tradeoffs between these goals would be set by the manager of the machine, and the adaptive system would strive to achieve these goals with even legacy applications.

As an example of node-local adaptation, we have recently focused on optimizing for performance, energy, and power in multisocket, multicore nodes, including NUMA nodes. This work has included adaptive paging mode selection [2], adaptive virtual core to physical core mapping [3], and page to memory bank mapping [1]. With virtual core mapping alone, we can increase the performance of the SPEC and PARSEC benchmarks by as much as 66%, reduce their energy by as much as 31%, and reduce their power by as much as 17%, with $< 0.05\%$ performance overhead for the monitoring and adaptation system itself.

As an example of machine-scale adaptation, we have been building a system that dynamically tracks memory content sharing across the machine [7]. Such information could be leveraged for numerous resiliency purposes. For example, it would facilitate more efficient checkpointing, or would let us add a selected degree of memory redundancy with a minimum additional memory footprint.

Both examples will operate with any guest OS and application.

Extensibility The second important property of the envisioned virtualization layer would be that it be *extensible*. The VMM, for example, could be augmented with desired new OS or virtual hardware functionality, allowing a guest that exploits a new execution model to run side-by-side with one that expects a traditional virtual hardware model. Similarly, the coordination between VMMs could be changed to better suit particular execution models. Particularly useful combinations of adaptation mechanisms/policies and extensions might prove to enhance the commercial prospects from the vendor perspective.

Because the VMM has ultimate control over the hardware, it can be used to manipulate, or construct, an application execution environment that extends that of the guest OS. As an example, the premise of our in-progress GEARS (Guest Examination and Revision Services) effort [4] is that the implementation of VMM-based services should be able to extend into the guest OS and application—to blur the line between the VMM and the application. GEARS currently allows us to transform the guest application (or OS) through system call interception and injection, kernel code injection, linking and invocation, and user code injection, linking, and injection.

2 Assessment

We now assess the proposed approach of an adaptive and extensible virtualization layer along the dimensions noted in the call for papers.

Challenges addressed The proposed approach would address the challenges of reliability, deeper memory hierarchies, and power outlined in the workshop charter. An adaptive system would directly be able to manipulate where and when VMs, virtual cores, and data are placed so as to achieve specific performance and power tradeoffs. Content tracking would serve as the basis for replication of data to enhance resilience, and VM snapshots would serve as the basis to allow replicated computation. Extensibility would permit the incremental deployment of new programming models and services while maintaining legacy compatibility. The virtualization layer in general might enhance the business case for exascale machines in a commercial environment.

Maturity Virtualization technologies have reached a high level of maturity in the commercial world, where VMMs underlie much of today's data center and cloud computing. The CPU and memory overheads of virtualization have long been known to be negligible. The primary issue has been I/O, which is of particular concern for high performance parallel computing, where communication plays such a central role. Self-virtualizing I/O devices and standards, such as IOMMUs (now common on Intel and AMD platforms) and the PCI SR-IOV standard, have resulted in a convergence of virtualized and native I/O performance. Within the HPC space itself, the V3VEE project (v3vee.org) has developed a publicly available VMM suitable for HPC [6], and demonstrated that extremely-low overhead virtualization at large scales for scientific workloads is entirely feasible [5]. A modern supercomputer could be run virtualized at all times, which would facilitate greater accessibility by allowing users to run arbitrary software stacks, while simultaneously providing a locus for adaptivity and extensibility.

Having virtualization in place would provide the playing field for long-term, sustainable innovation in the adaptivity and extensibility aspects of the approach. Adaptive (or autonomic) computing and extensible operating systems have a long history outside of HPC.

Uniqueness The proposed approach invokes similar approaches being taken commercially, particularly in data center and cloud computing. However, while initial results are promising, we simply do not yet know whether the approach would work for exascale scientific computing. At this point, the focus of other systems research and development programs appears to be elsewhere. If we believe that the proposed approach has potential, and that exascale scientific computing will not simply converge with data center computing, then it it worth pursuing.

Novelty While virtualization, adaptive systems, and extensible operating systems are well attested in the literature and commercially, the approach is a novel one for exascale HPC systems, as noted above. An important novel aspect of this approach is that it will enable further ongoing innovation by its very nature.

Applicability While there are reasons to believe otherwise, it is conceivable that data center and supercomputer hardware will see increasing convergence. If this is the case, then the proposed approach would help the scientific community to better leverage future data center-based computing. The effort will clearly identify the properties that an virtualization layer, which the data center will surely have, will require to be successful for scientific computing.

Effort On the development side, requiring that vendors include an open virtualization layer does not appear to onerous on its face, as numerous options are available already. On the research side, the approach would use existing X-Stack research as a springboard. The research effort is difficult to quantify as it would depend on the specific adaptivity and extensibility techniques to be explored. The likely scale would be similar to existing medium-size X-Stack projects.

References

[1] BAE, C., AND JAMAL, T. Energy-aware memory management through database buffer management. In *Proceedings of the Third Workshop on Energy Efficient Design (WEED 2011)* (June 2011).

[2] BAE, C., LANGE, J., AND DINDA, P. Enhancing virtualized application performance through dynamic adaptive paging mode selection. In *Proceedings of the 8th International Conference on Autonomic Computing (ICAC 2011)* (June 2011).

[3] BAE, C., XIA, L., DINDA, P., AND LANGE, J. Adaptive virtual core mapping to improve power, energy, and performance in multi-socket multicores. In *Proceedings of the 21st ACM Symposium on High-performance Parallel and Distributed Computing (HPDC 2012)* (June 2012).

[4] HALE, K., XIA, L., AND DINDA, P. Shifting GEARS to enable guest-context virtual services. In *Proceedings of the 9th International Conference on Autonomic Computing (ICAC 2012)* (September 2012).

[5] LANGE, J., PEDRETTI, K., DINDA, P., BRIDGES, P., BAE, C., SOLTERO, P., AND MERRITT, A. Minimal overhead virtualization of a large scale supercomputer. In *Proceedings of the 2011 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE 2011)* (March 2011).

[6] LANGE, J., PEDRETTI, K., HUDSON, T., DINDA, P., CUI, Z., XIA, L., BRIDGES, P., GOCKE, A., JACONETTE, S., LEVENHAGEN, M., AND BRIGHTWELL, R. Palacios and kitten: New high performance operating systems for scalable virtualized and native supercomputing. In *Proceedings of the 24th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2010)* (April 2010).

[7] XIA, L., AND DINDA, P. A case for tracking and exploiting memory content sharing in virtualized large-scale parallel systems. In *Proceedings of the 6th International Workshop on Virtualization Technologies in Distributed Computing (VTDC 2012)* (June 2012).